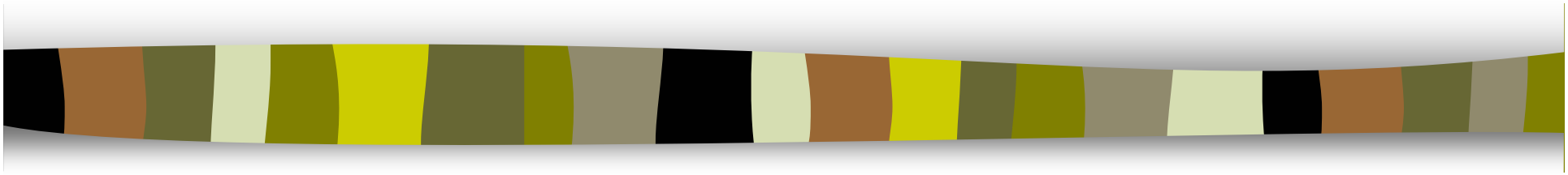
Curso de PHP



Tema 4: Acceso a bases de datos
MySQL en PHP

José Mariano González Romano
mariano@lsi.us.es

Tema 4: Acceso a bases de datos MySQL en PHP

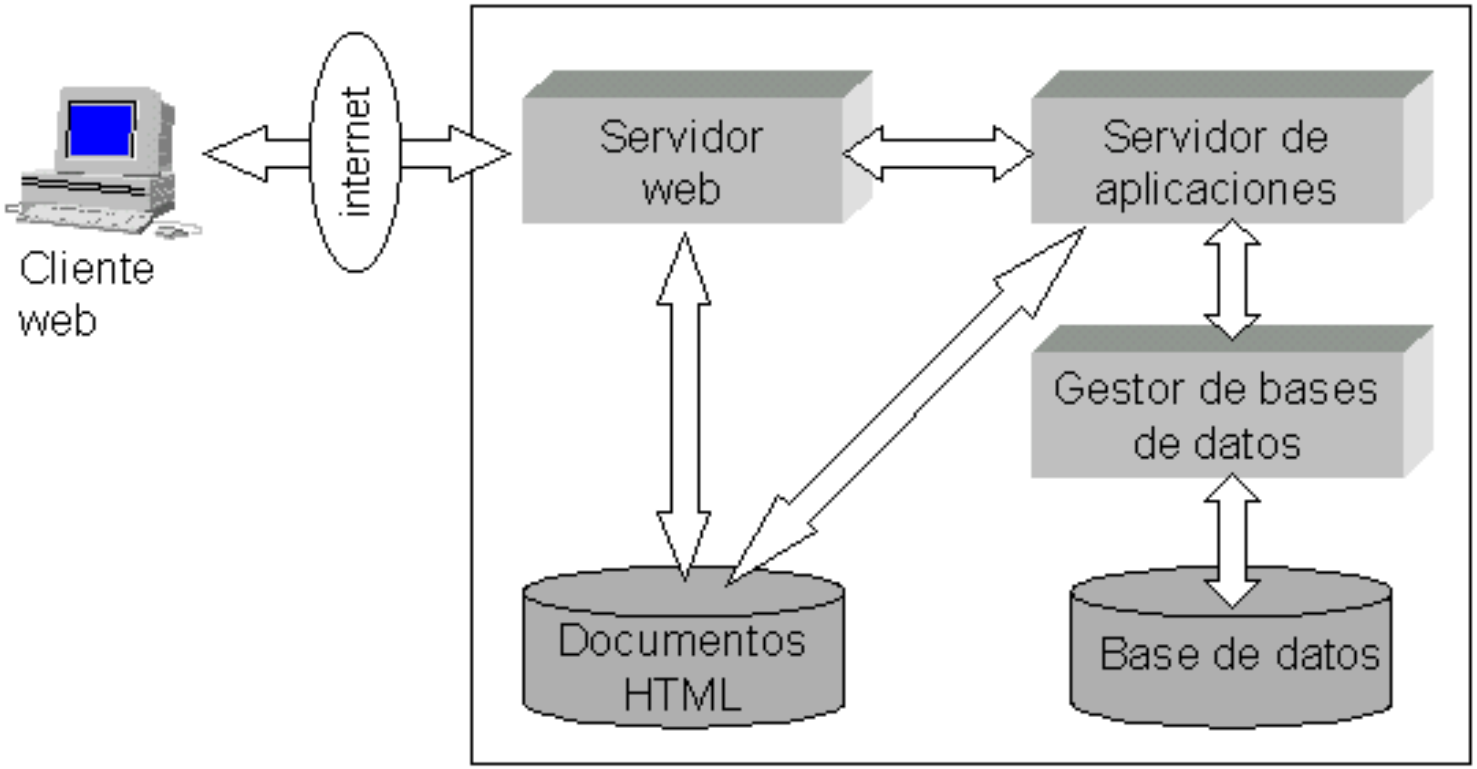
1. Bases de datos en la Web
2. Instalación y configuración de MySQL
3. MySQL
4. Herramientas de administración: phpMyAdmin
5. Lenguaje SQL
6. Funciones de PHP para el acceso a bases de datos MySQL
7. Ejercicios
8. Consulta avanzada de tablas

Bases de datos en la Web

- Las bases de datos permiten almacenar de una forma estructurada y eficiente toda la información de un sitio web
- Ventajas
  - Proporcionar información actualizada
  - Facilitar la realización de búsquedas
  - Disminuir los costes de mantenimiento
  - Implementar sistemas de control de acceso
  - Almacenar preferencias de los usuarios

Bases de datos en la Web

- Esquema básico de un sitio web soportado por bases de datos:



Instalación y config. de MySQL

- Pasos:
  - Descargar
  - Descomprimir e instalar
  - Configurar
  - Arrancar
  - Conectar con el servidor
  - Instalar la extensión para MySQL de PHP

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 1: descargar**
  - Conectarse a la página web de MySQL, www.mysql.com
  - Seleccionar la pestaña *Developer Zone* y dentro de ella la sección *Downloads*
  - Elegir la versión estable más reciente de *MySQL Community Edition – Database server and client*. A fecha de febrero de 2006 es la 5.0
  - Seleccionar la plataforma Windows y descargar el archivo comprimido a una carpeta temporal

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 2: descomprimir e instalar**
  - Descomprimir el archivo descargado en una carpeta temporal
  - Ejecutar el archivo setup.exe y seguir las indicaciones:
    - Seleccionar instalación típica
    - Aceptar la carpeta de instalación por defecto
    - Saltar el formulario de registro
  - Pulsar el botón *Finish* para terminar la instalación y pasar a la configuración del servidor

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 3: configurar**
  - Al finalizar la instalación se ejecuta el asistente para la configuración del servidor. Seguir sus indicaciones:
    - Elegir la configuración estándar
    - Seleccionar la instalación como servicio Windows y Marcar la casilla para lanzar el servidor automáticamente al arrancar el sistema
    - Establecer una contraseña para el administrador (root)
    - Pulsar el botón *Execute* para realizar la configuración
    - Pulsar el botón *Finish* para finalizar el asistente

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 4: arrancar**
  - Si se indicó el arranque automático en la configuración, el servidor se inicia de forma automática al arrancar el sistema
  - En caso contrario hay que iniciarlo manualmente con Inicio > Programas > MySQL > MySQL Server 5.0 > MySQL Server Instance Config Wizard

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 5: conectar**
  - Formas de establecer la conexión con el servidor:
    - Desde la línea de órdenes con Inicio > Programas > MySQL > MySQL Server 5.0 > MySQL Command Line Client

```
Enter password: ********
mysql>
…
mysql> exit
```

    - Mediante alguna herramienta que proporcione una interfaz gráfica como phpMyAdmin
    - Desde una página web mediante la interfaz que proporciona MySQL. Es lo que haremos con la biblioteca de funciones de MySQL que posee PHP

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 5: conectar**
  - Para conectar con el servidor hay que crear antes una cuenta de usuario y asignarle los correspondientes permisos de acceso. En general para una base de datos es conveniente definir al menos dos usuarios:
    - Un usuario anónimo que tenga permisos de lectura sobre las tablas que se estime adecuado
    - Un usuario administrador que tenga permisos para insertar, modificar o eliminar elementos de las tablas de la base de datos
  - En cada conexión hay que indicar el nombre del usuario, su contraseña y la máquina desde la que se realiza la conexión (localhost si es la propia máquina donde reside el servidor, que es lo habitual en el acceso desde la Web)

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 6: MySQL-PHP**
  - En PHP 5 no se instala por defecto la extensión para el acceso a bases de datos MySQL y hay que hacerlo manualmente
  - Se hace de la siguiente manera:
    - Editar el fichero php.ini y habilitar la extensión:

      ```
      extension=php_mysql.dll
      ```

    - Añadir la ruta c:\php al PATH del sistema (ver las instrucciones para hacerlo en el fichero install.txt que hay en la carpeta c:\php)

MySQL

- **Características de MySQL**
  - Modelo relacional, multiusuario
- **Tipos de datos**
  - Numéricos
    - tinyint, smallint, mediumint, int, integer, bigint
    - decimal, float, numeric
  - Fecha y hora
    - date, time, datetime, year, timestamp
  - Cadena
    - char, varchar
    - tinytext, text, mediumtext, longtext
    - tinyblob, blob, mediumblob, longblob
    - enum, set
  - Debe elegirse adecuadamente el tipo y el tamaño de cada campo

MySQL

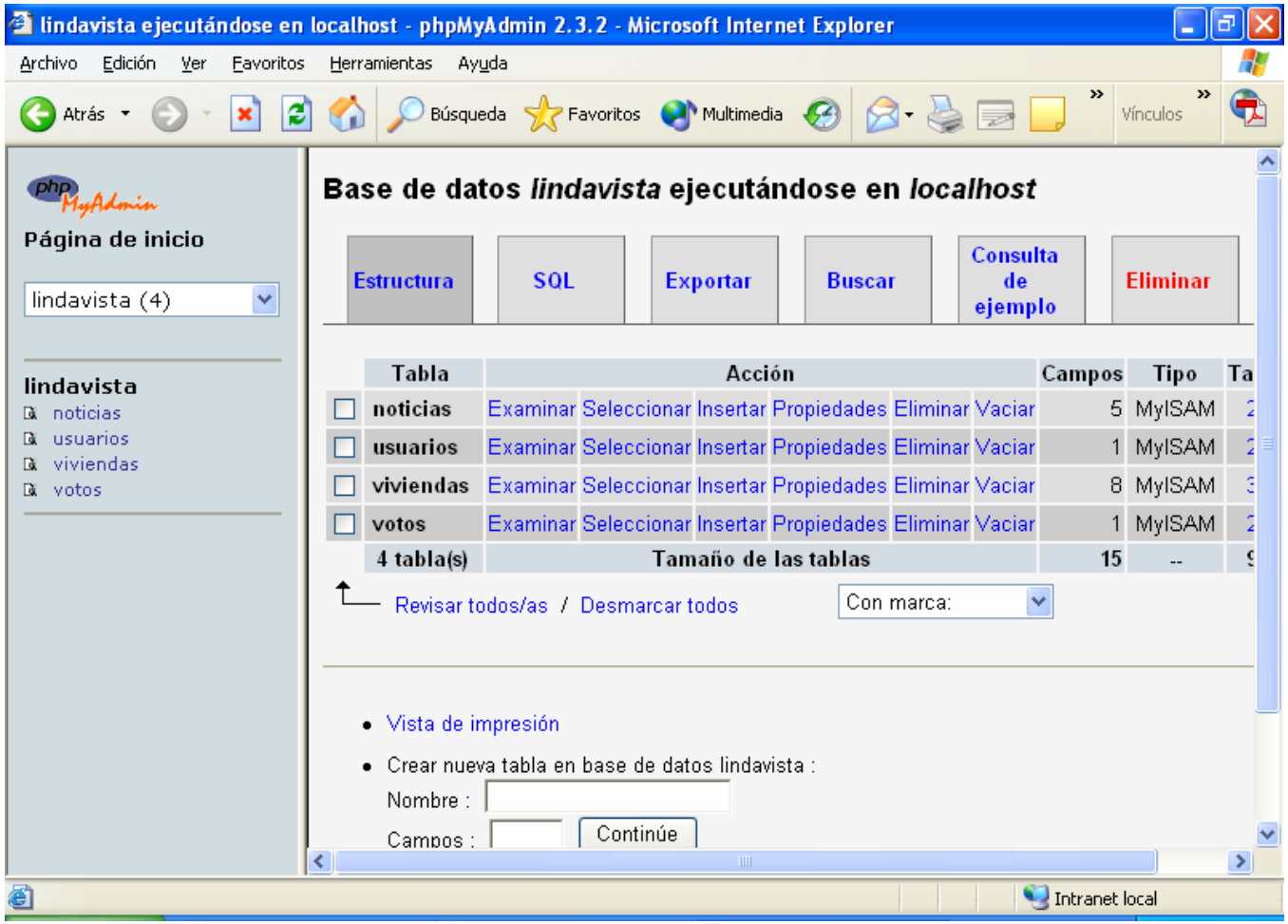
- **Operadores**
  - Aritméticos
    - +, -, *, /
  - Comparación
    - =, !=, <=, <, >=, >, IS NULL, IS NOT NULL
  - Lógicos
    - not (!), and (&&), or (||), xor
- **Funciones**
  - Funciones de cadena
  - Funciones de comparación de cadenas
  - Funciones numéricas
  - Funciones de fecha y hora
  - Funciones de agregado

Herramientas de administración: phpMyAdmin

- **phpMyAdmin** es una herramienta para la administración del servidor de bases de datos MySQL
- Dispone de una interfaz gráfica y es de libre distribución
- Permite realizar todo tipo de operaciones sobre bases de datos:
  - crear, borrar y modificar tablas
  - consultar, insertar, modificar y eliminar datos
  - definir usuarios y asignar permisos
  - realizar copias de seguridad
  - etc
- Está escrita en php y se ejecuta desde el navegador
- Si está instalada en la carpeta phpmyadmin, se ejecuta escribiendo en la barra de direcciones del navegador la url

  http://localhost/phpmyadmin/
- Puede administrar bases de datos locales y remotas

phpMyAdmin


lindavista ejecutándose en localhost - phpMyAdmin 2.3.2 - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia Vínculos
Página de inicio
lindavista (4)
lindavista
noticias
usuarios
viviendas
votos
Base de datos lindavista ejecutándose en localhost
Estructura SQL Exportar Buscar Consulta de ejemplo Eliminar
Tabla Acción Campos Tipo
noticias Examinar Seleccionar Insertar Propiedades Eliminar Vaciar 5 MyISAM
usuarios Examinar Seleccionar Insertar Propiedades Eliminar Vaciar 1 MyISAM
viviendas Examinar Seleccionar Insertar Propiedades Eliminar Vaciar 8 MyISAM
votos Examinar Seleccionar Insertar Propiedades Eliminar Vaciar 1 MyISAM
4 tabla(s) Tamaño de las tablas 15 --
Revisar todos/as / Desmarcar todos
Con marca:
Vista de impresión
Crear nueva tabla en base de datos lindavista :
Nombre :
Campos :
Continúe
Intranet local

phpMyAdmin

- Pasos para su instalación:
  - Descargar
  - Descomprimir
  - Configurar
  - Ejecutar

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 1: descargar**
  - Conectarse a la dirección http://www.phpmyadmin.net
  - Seleccionar *Downloads*
  - Seleccionar la versión más reciente. A fecha de febrero de 2006 es la 2.7.0
  - Seleccionar un mirror e iniciar la descarga

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 2: descomprimir**
  - Descomprimir debajo de la carpeta raíz de la web, c:\htdocs
  - Cambiar el nombre de la carpeta creada (phpMyAdmin-2.7.0-pl2) a phpmyadmin

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 3: configurar**
  - El fichero de configuración se llama config.inc.php
  - Este fichero no existe sino que hay que crearlo. Para ello se hace una copia del fichero config.default.php, situado en la carpeta donde se haya instalado phpMyAdmin, y se le pone como nombre config.inc.php
  - Seguidamente se modifica el fichero config.inc.php
  - Configuración típica para un servidor local:

```
...
$cfg['Servers'][$i]['host']  = 'localhost'; //MySQL hostname
$cfg['Servers'][$i]['user']  = 'root';      //MySQL user
$cfg['Servers'][$i]['password'] = 'clave';  //MySQL password
...
```

    siendo ‘clave’ la contraseña asignada al administrador (root) de MySQL durante su instalación

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 4: ejecutar**
  - Ejecutar Apache
  - Abrir el navegador y teclear la url http://localhost/phpmyadmin/index.php

Lenguaje SQL

- SQL (*Structured Query Language*) es el lenguaje que se utiliza para comunicarse con la base de datos
- Procedimiento de comunicación con la base de datos:

Lenguaje SQL

- Las instrucciones más habituales son SELECT, INSERT, UPDATE, DELETE
- Veamos su sintaxis básica y algunos ejemplos de uso
- Para ello utilizaremos una tabla **noticias** con cinco campos: un identificador único de la noticia, el título de la noticia, el texto de la noticia, la categoría de la noticia y la fecha de publicación de la noticia

noticias
id
título
texto
categoría
fecha

Lenguaje SQL

- SELECT

Sintaxis:

```
SELECT expresión FROM tabla
[WHERE condición]
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC] ,...]
[LIMIT [offset,] row_count | row_count OFFSET offset]
```

Ejemplo:

```
SELECT * from noticias WHERE fecha=CURDATE() LIMIT 10 ORDER BY fecha
    DESC
```

Obtiene las noticias del día con un tope máximo de 10, ordenadas de la más reciente a la más antigua

Lenguaje SQL

- INSERT

Sintaxis:

```
INSERT [INTO] nombre_tabla [(nombre_columna,...)]
VALUES ((expresión | DEFAULT),...), (...),...
INSERT [INTO] nombre_tabla
SET nombre_columna=(expresión | DEFAULT), ...
```

Ejemplo:

```
INSERT INTO noticias (id, titulo, texto, categoria, fecha) VALUES
   (37, “Nueva promoción en Nervión”, “145 viviendas de lujo en
   urbanización ajardinada situadas en un entorno privilegiado”,
   “promociones”, CURDATE())
```

Inserta una noticia con los valores indicados

Lenguaje SQL

- UPDATE

Sintaxis:

```
UPDATE nombre_tabla
SET nombre_columna1=expr1 [, nombre_columna2=expr2 ...]
[WHERE condición]
[ORDER BY ...]
[LIMIT row_count]
```

Ejemplo:

```
UPDATE noticias SET categoria = “ofertas” WHERE id=37
```

Modifica la categoría de la noticia con id=37 de la tabla

Lenguaje SQL

- DELETE

Sintaxis:

```
DELETE FROM nombre_tabla
[WHERE condición]
[ORDER BY ...]
[LIMIT row_count]
```

Ejemplo:

```
DELETE FROM noticias WHERE fecha < CURDATE()-10
```

Borra las noticias con más de 10 días de antigüedad

Funciones de PHP para el acceso a bases de datos MySQL

- Los pasos para acceder desde PHP a una base de datos son los siguientes:
  - Conectar con el servidor de bases de datos
  - Seleccionar una base de datos
  - Enviar la instrucción SQL a la base de datos
  - Obtener y procesar los resultados
  - Cerrar la conexión con el servidor de bases de datos

Acceso a bases de datos MySQL

- Las funciones concretas de MySQL que realizan estas operaciones son:
  - Conectar con el servidor de bases de datos:
    - **mysql_connect()**
  - Seleccionar una base de datos:
    - **mysql_select_db()**
  - Enviar la instrucción SQL a la base de datos:
    - **mysql_query()**
  - Obtener y procesar los resultados:
    - **mysql_num_rows() y mysql_fetch_array()**
  - Cerrar la conexión con el servidor de bases de datos:
    - **mysql_close()**

Acceso a bases de datos MySQL

- Conectar con el servidor de bases de datos: **mysql_connect()**
  - Devuelve un identificador de la conexión en caso de éxito y false en caso contrario
- Sintaxis:

```
$conexion = mysql_connect (servidor, username, password);
```

- Ejemplo:

```
$conexion = mysql_connect (“localhost”, “cursophp”, “”)
   or die (“No se puede conectar con el servidor”);
$conexion = mysql_connect (“localhost”, “cursophp-ad”, “php.hph”)
   or die (“No se puede conectar con el servidor”);
```

Acceso a bases de datos MySQL

- Seleccionar una base de datos: **mysql_select_db()**
  - Devuelve true en caso de éxito y false en caso contrario
- Sintaxis:

```
mysql_select_db (database);
```

- Ejemplo:

```
mysql_select_db (“lindavista”)
   or die (“No se puede seleccionar la base de datos”);
```

Acceso a bases de datos MySQL

- Enviar la instrucción SQL a la base de datos: **mysql_query()**
  - Devuelve un identificador o true (dependiendo de la instrucción) si la instrucción se ejecuta correctamente y false en caso contrario
- Sintaxis:

```
$consulta = mysql_query (instrucción, $conexion);
```

- Ejemplo:

```
$consulta = mysql_query (“select * from noticias”, $conexion)
   or die (“Fallo en la consulta”);
```

Acceso a bases de datos MySQL

- Obtener y procesar los resultados: **mysql_num_rows(), mysql_fetch_array()**
  - En el caso de que la instrucción enviada produzca unos resultados, mysql_query() devuelve las filas de la tabla afectadas por la instrucción
  - mysql_num_rows() devuelve el número de filas afectadas
  - Para obtener las distintas filas del resultado se utiliza la función mysql_fetch_array(), que obtiene una fila del resultado en un array asociativo cada vez que se invoca

- Sintaxis:

```
$nfilas = mysql_num_rows ($consulta);
$fila = mysql_fetch_array ($consulta);
```

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

Instrucción:

```
select * from noticias where categoria=“promociones”
```

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

`$consulta`

`$nfilas=3`

Instrucción:

```
select * from noticias where categoria=“promociones”
```

Acceso a bases de datos MySQL

- Obtención de las filas:

```
$nfilas = mysql_num_rows ($consulta);
if ($nfilas > 0)
{
   for ($i=0; $i<$nfilas; $i++)
   {
      $fila = mysql_fetch_array ($consulta);
      procesar fila i-ésima de los resultados
   }
}
```

Acceso a bases de datos MySQL

- Obtener los resultados: **mysql_num_rows(), mysql_fetch_array()**
  - Para acceder a un campo determinado de una fila se usa la siguiente sintaxis:

```
$fila[“nombre_campo”]// por ser un array asociativo
$fila[$i]            // $i=índice del campo desde 0
```

- Ejemplo:

```
for ($i=0; $i<$nfilas; $i++)
{
   $fila = mysql_fetch_array ($consulta);
   print “Título: “ . $fila[“titulo”];
   print “Fecha: “ . $fila[“fecha”];
}
```

Acceso a bases de datos MySQL

- Cerrar la conexión con el servidor de bases de datos: **mysql_close()**
- Sintaxis:

```
mysql_close ($conexion);
```

- Ejemplo

```
mysql_close ($conexion);
```

Ejercicios

- **Ejercicio 1: consulta de noticias**
  - Ejercicio simple de acceso a una tabla de una base de datos. Ilustra cómo conectar con una base de datos, enviar una consulta, recuperar los resultados y mostrarlos en pantalla.
  - Pasos previos:
    1. Crear la base de datos lindavista
    2. Crear la tabla noticias en la base de datos lindavista
    3. Crear el usuario ‘cursophp’ (sin contraseña)
    4. Dar al usuario ‘cursophp’ permiso SELECT sobre la tabla noticias de la base de datos lindavista

Ejercicios

- **Ejercicio 2: encuesta**
  - Ilustra cómo actualizar una tabla y cómo generar gráficos simples
  - Pasos previos:
    1. Crear la tabla votos en la base de datos lindavista
    2. Dar al usuario ‘cursophp’ permisos SELECT y UPDATE sobre la tabla votos de la base de datos lindavista

Ejercicios

- **Ejercicio 3: inserción de noticias**
  - Se creará un formulario para insertar las noticias en la base de datos
  - Pasos previos:
    1. Crear el usuario cursophp-ad (contraseña: php.hph)
    2. Dar al usuario ‘cursophp-ad’ permisos SELECT e INSERT sobre la tabla noticias de la base de datos lindavista

Ejercicios

- **Ejercicio 4: eliminación de noticias**
  - Se creará un formulario para eliminar noticias de la base de datos
  - Primero se seleccionarán las noticias a eliminar de entre todas las existentes y a continuación se eliminarán
  - Paso previo:
    1. Dar al usuario ‘cursophp-ad’ permiso DELETE sobre la tabla noticias de la base de datos lindavista

Consulta avanzada de tablas

- Objetivo:
  - mostrar los resultados de la consulta **divididos en bloques de un número determinado de elementos** (por ejemplo, de 5 en 5)
- Requisitos:
  - Recuperar un número limitado de elementos de la tabla
  - Implementar un mecanismo de navegación que permita avanzar al siguiente o volver al anterior bloque de elementos

Consulta avanzada de tablas

- Para recuperar un número fijo de elementos de una tabla se utiliza la opción **LIMIT** de la orden SELECT. Así, por ejemplo,

```
SELECT * from noticias LIMIT 0, 5
```

recupera los 5 primeros elementos de la tabla. Y en general,

```
SELECT * from noticias LIMIT $comienzo, $num
```

recupera $num elementos a partir de la posición $comienzo

Consulta avanzada de tablas

- La variable $num tendrá un valor constante (en este caso 5), mientras que la variable $comienzo se incrementará o decrementará en 5 unidades al pasar a la página siguiente o anterior
- Para ello se pasará la variable como parámetro en el enlace asociado al botón correspondiente
- Por ejemplo, el código para el botón siguiente será:

```
"<A HREF='" . $_SERVER['PHP_SELF'] . "?comienzo=" .
($comienzo + $num) . "'>Siguiente</A>"
```

- Habrá que comprobar previamente que el nuevo valor de comienzo es válido, es decir, que se encuentra dentro de los límites de la tabla devuelta por la consulta

Consulta avanzada de tablas

- **Ejercicio 5: consulta de noticias con paginado de resultados**
  - Ejercicio que ilustra cómo dividir los resultados de una consulta en varias páginas y permitir desplazarse entre ellas

Consulta avanzada de tablas

- Objetivo:
  - Mostrar los resultados de una consulta de manera que se puedan **filtrar en función del valor de una determinada columna** de la tabla
- Requisitos:
  - Recuperar de una tabla los elementos que cumplan una determinada condición
  - Permitir seleccionar un valor de entre los valores posibles de una columna

Consulta avanzada de tablas

- Para recuperar los elementos de una tabla que cumplen una condición se utiliza la opción **WHERE** de la orden SELECT. Por ejemplo,

```
SELECT * from noticias WHERE categoria=’ofertas’
```

  recupera las noticias cuya categoría tiene el valor “ofertas”. Y en general,

```
SELECT * from noticias WHERE categoria =’$categoria’
```

  recupera las noticias cuya categoría tiene el valor dado por la variable $categoria

- La variable $categoria se obtendrá de un elemento SELECT

Consulta avanzada de tablas

- **Ejercicio 6: consulta de noticias con filtrado de resultados**
  - Ejercicio que ilustra cómo mostrar los resultados de una consulta filtrados en función del valor de una columna de tipo enumerado con actualización manual al pulsar un botón

Consulta avanzada de tablas

- Objetivo:
  - Modificar el ejercicio anterior para que los resultados se actualicen de forma automática al seleccionar un nuevo valor para la columna
- Requisitos:
  - Obtener automáticamente los valores de una columna de tipo enumerado a partir de la tabla
  - Utilizar código JavaScript para detectar un cambio en la opción seleccionada de un elemento de tipo SELECT

Consulta avanzada de tablas

- Función JavaScript que actualiza una página en función de la opción seleccionada en el elemento SELECT de nombre ‘categoria’ del formulario de nombre ‘selecciona’:

```
<SCRIPT LANGUAGE='JavaScript'>
<!--
function actualizaPagina ()
{
   i = document.forms.selecciona.categoria.selectedIndex;
   categoria =
     document.forms.selecciona.categoria.options[i].value;
   window.location = ‘muestra_noticias.php?categoria=' +
     categoria;
}
// -->
</SCRIPT>
```

Consulta avanzada de tablas

- Para ejecutar la función JavaScript es preciso asociarla al elemento SELECT a través del evento ONCHANGE, que se activa cuando se modifica la opción seleccionada:

```
<SELECT NAME='categoria' ONCHANGE='actualizaPagina()'>
```

Consulta avanzada de tablas

- Es conveniente que la opción seleccionada aparezca marcada por defecto al actualizar la página
- Para ello es preciso generar las opciones del elemento SELECT de forma automática
- De esta manera se puede comparar cada una de las opciones con el valor proveniente del formulario y colocar el atributo SELECTED a la que corresponda

Consulta avanzada de tablas

- En el formulario del ejercicio anterior los valores de los campos de tipo enumerado (ENUM) están escritos directamente en el propio código. Así, para la categoría de noticia se tiene:

```
<SELECT NAME=“categoria">
   <OPTION VALUE=“promociones">promociones
   <OPTION VALUE=“ofertas" SELECTED>ofertas
   <OPTION VALUE=“costas">costas
</SELECT>
```

- Este hecho no es muy deseable ya que si se modifican los valores de la categoría en la tabla hay que modificar también el código del formulario

Consulta avanzada de tablas

- Lo ideal es tomar los valores automáticamente de la tabla en lugar de escribirlos ‘a mano’ en el código. Para ello se utiliza la siguiente instrucción SQL:

```
SHOW columns FROM noticias LIKE ‘categoria‘
```

que devuelve una tabla con las propiedades del campo ‘categoria’. El elemento [1] de esta tabla contiene lo siguiente:

```
enum(“promociones",“ofertas",“costas")
```

- A partir de esta información podemos obtener los valores del tipo enumerado y generar las opciones del elemento SELECT. Sólo queda marcar con SELECTED la opción que estaba seleccionada en caso de haberse enviado el formulario

Consulta avanzada de tablas

- **Ejercicio 7: consulta de noticias con filtrado de resultados**
  - Ejercicio que ilustra cómo mostrar los resultados de una consulta filtrados en función del valor de una columna de tipo enumerado con actualización automática a través de JavaScript